UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/817,437	03/26/2001	Stephen J. Todd	JW-EMC-007	9135

24227	7590	12/15/2005

EMC CORPORATION
OFFICE OF THE GENERAL COUNSEL
176 SOUTH STREET
HOPKINTON, MA 01748

EXAMINER
VO, TED T

ART UNIT	PAPER NUMBER
2191	

DATE MAILED: 12/15/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

PTO-90C (Rev. 10/03)

	Office Action Summary	Application No.	Applicant(s)
		09/817,437	TODD ET AL.
		Examiner	Art Unit
		Ted T. Vo	2191

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE <u>3</u> MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.
- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *19 September 2005*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-89* is/are pending in the application.

    4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-89* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *02 August 2004* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

    Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

    Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

    a)☐ All b)☐ Some * c)☐ None of:

      1.☐ Certified copies of the priority documents have been received.

      2.☐ Certified copies of the priority documents have been received in Application No. _____.

      3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

    * See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.

4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1.   This action is in response to the amendment filed on 09/19/2005.

Claims 1-89 are pending in the application.

Response to Arguments

2.   Applicants' arguments have been fully considered, but are moot in view of a new ground of

rejection.

Claim Rejections - 35 USC § 102

3.   The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for

the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or
in public use or on sale in this country, more than one year prior to the date of application for
patent in the United States.

4.   Claims 1-89 are rejected under 35 U.S.C. 102(b) as being anticipated by Chen et al., "A C++

Data Model Supporting Reachability Analysis and Dead Code Detection", 1998.

Given the broadest reasonable interpretation of followed claims in light of the specification.

As per Claim 1:

Note: The specification refers CIM or XML as a different computer language from the first computer

language such as C++. However, the definition of CIM is "Common Information Model", and XML

document file that is viewable by a browser.

Accordingly, Chen discloses,

A computer system employing management software written in a first computer language compatible with first software architecture and not compatible with second software architecture, said system comprising:

a schema formed within said first software architecture (See p. 690, left col., first bullet);

header files contained within said schema, said header files being represented in said first language and capable of being utilized by said management software (refer to C++ code mapped to the C++ data model, e.g. see p. 683, left col., A C++ DATA MODEL);

means for manipulating said header files to locate public functions and/or data attributes of said header files; (See p. 683, left col., A C++ DATA MODEL: Note: with a presence of a user and a program C++, it has means of *manipulating said header files to locate public functions and/or data attributes of said header files*);

means, responsive to operation of said manipulating means, for emitting code that calls said public functions and/or data attributes in said first language to obtain called public functions and/or data attributes; (See p. 683, left col., A C++ DATA MODEL) *and,*

means for converting said called public functions and/or data attributes to representations of said called public functions and/or data attributes formed in a different computer language compatible with said second software architecture (See p. 683, left col., A C++ DATA MODEL; and see p. 690, left col. Sec. B, 'using an instance compiler that takes a specification file for a new language or document type and generates the complete query and visualization environment automatically').

Chen teaches converting existing legacy code such as C++ (First computer language) into a specification file of object mode. Similarly to HTML/XML document file, the specification file is visualized by CIAO. For example, see in p. 683, Chen's discussion of C++ source file containing a header file and a Model that corresponds to C++ source file. See in p. 690, Chen's discussion of a document file converted from the C++ source file.

<u>As per Claim 2</u>: Chen discloses, *The computer system of claim 1 further comprising means for forwarding said representations to desired destinations within and beyond said system* (See Abstract, Introduction; refer to the location that stores software repository/existing legacy code, relational database).

As per Claim 3: Chen discloses, *The computer system of claim 1 and wherein said first computer language is RAID++* (Equivalent Functionality: C++ as mentioned) *and said different computer language is XML/CIM* (Equivalent Functionality: the specification file that is viewable by CIAO as mentioned).

As per Claim 4: Chen discloses, *The computer system of claim 1 and wherein said first computer language is an object-oriented language defining computer data and commands as objects, said manipulating means comprising: means for opening at least one of said header files containing a declaration of at least one of said objects; means for parsing said at least one of said header files to obtain name of class and name of parent class to which said at least one of said objects belongs; and, means for creating a subroutine for accepting said at least one of said objects in said first computer language and generating the equivalent of said at least one of said objects in a different computer language compatible with said second software architecture.* (See p. 683, refer to the term C++ (object/first computer language) and C++ Data model/specification file ('*a different computer language compatible with said second software architecture*', and see p. 690, instance compiler).

As per Claim 5: Chen discloses, *The computer system of claim 1 further comprising means for inhibiting initiation of operation of said converting means until said public functions and/or data attributes of said header files are located* (See discussion in p. 683, A C++ DATA MODEL).

As per Claim 6: Chen discloses, *The computer system of claim 1 further comprising means for initiating operation of said converting means upon locating the first of any one of said public functions and/or data attributes.* (See discussion in p. 683, A C++ DATA MODEL).

As per Claim 7: Chen discloses, *The computer system of claim 1and wherein said first computer language is C++ and said different computer language is XML/CIM* (Equivalent Functionality: the specification file that is viewable by CIAO as mentioned).

As per Claim 8: Chen discloses, *The computer system of claim 1 and wherein said first computer language is a first object-oriented language* (Refer to C++); *capable of pictorial representation typically in a parent-child tree configuration and said different computer language is a second object oriented language capable of pictorial representation typically in a flat database configuration* (Refer to C++ DATA MODEL).

As per Claim 9: Chen discloses, *The computer system of claim 1 further comprising means for inhibiting initiation of operation of said converting means until said public functions and/or data attributes of at least one of said header files are located* (See discussion in p. 683, A C++ DATA MODEL).

As per Claim 10: Chen discloses, *The computer system of claim 1 and wherein said management software is storage management software* (See Whole text, Chen's C++ with relational database is storage management software).

As per Claim 11: Chen discloses, *The computer system of claim 1 and wherein said management software is selected from the group consisting of storage, printer, and server management software* (See Whole text, Chen's C++ is dealt with software computer related).

As per Claim 12: Chen discloses, *A computer network employing a computer system utilizing management software written in a first computer language compatible with first software architecture and not compatible with second software architecture, said network comprising:*

a schema formed within said first software architecture;

header files contained within said schema, said header files being represented in said first language and capable of being utilized by said management software;

apparatus for manipulating said header files to locate public functions and/or data attributes of said header files; and,

apparatus, responsive to operation of said manipulating apparatus, for emitting code that calls said public functions and/or data attributes in said first language to obtain called public functions and/or data attributes and that converts said called public functions and/or data attributes to representations of said called public functions and/or data attributes formed in a different computer language compatible with said second software architecture.

Same rejection as referred to the rationale addressed in As per Claim 1.

As per Claim 13: Chen discloses, *The network of claim 12 and wherein said computer system further comprises apparatus for forwarding said representations to desired destinations within and outside of said network* (Refer to the location that stores software repository/existing legacy code).

As per Claim 14: Chen discloses, *The network of claim 12 and wherein said first computer language is*

RAID++ and said different computer language is XML/CIM.

Same rejection as referred to the rationale addressed in As per Claim 3.

<u>As per Claim 15</u>: Chen discloses, *The network of claim 12 and wherein said first computer language is an object-oriented language defining computer data and commands as objects, said manipulating apparatus comprising: apparatus for opening one of said header files containing a declaration of one of said objects; apparatus for parsing said one of said header files to obtain name of class and name of parent class to which said one of said objects belongs; and, apparatus for creating a subroutine for accepting said one of said objects in said first computer language and generating the equivalent of said one of said objects in a different computer language compatible with said second software architecture.*

Same rejection as referred to the rationale addressed in As per Claim 4.

<u>As per Claim 16</u>: Chen discloses, *The network of claim 12 further comprising apparatus for inhibiting initiation of operation of said converting apparatus until said public functions and/or data attributes of said header files are located.*

Same rejection as referred to the rationale addressed in As per Claim 5.

<u>As per Claim 17</u>: Chen discloses, *The network of claim 12 further comprising apparatus for initiating operation of said converting apparatus upon locating the first of any one of said public functions and/or data attributes.*

Same rejection as referred to the rationale addressed in As per Claim 6.

<u>As per Claim 18</u>: Chen discloses, *The network of claim 12 and wherein said first computer language is C++ and said different computer language is XML/CIM.*

Same rejection as referred to the rationale addressed in As per Claim 7.

<u>As per Claim 19</u>: Chen discloses, *The network of claim 12 and wherein said first computer language is a first object-oriented language capable of pictorial representation typically in a parent child tree configuration and said different computer language is a second object-oriented language capable of pictorial representation typically in a flat database configuration.*

Same rejection as referred to the rationale addressed in As per Claim 8.

<u>As per Claim 20</u>: Chen discloses, *The network of claim 12 further comprising apparatus for inhibiting initiation of operation of said converting apparatus until said public functions and/or and data attributes of at least one of said header files are located.*

Same rejection as referred to the rationale addressed in As per Claim 9.

<u>As per Claim 21</u>: Chen discloses, *The network of claim 12 further comprising a SAN which communicates with and is controlled by said computer system.* Refer to Repository, existing legacy code.

<u>As per Claim 22</u>: Chen discloses, *The network of claim 12 and wherein said management software is storage management software.* (See Abstract and Introduction; refer to Repository, existing legacy code).

<u>As per Claim 23</u>: Chen discloses, *The network of claim 12 and wherein said management software is selected from the group consisting of storage, printer, and server management software.*

Same rejection as referred to the rationale addressed in As per Claim 11.

<u>As per Claim 24</u>: Chen discloses, *A method for utilizing standardized software architecture to be practiced in a computer system employing management software written in a first computer language compatible with first software architecture and not compatible with said standardized software architecture* (Refer to the computer that visualizes the Object Model, For example in the section VI, Chen refers the Object model is visualized under Netscape source code, HTML, XML, where the refer the computer and C++ software as *first computer language compatible with first software architecture), said method comprising:*

said management software utilizing a schema having header files in said first language;

manipulating said header files to locate public functions and/or data attributes of said header files; and,

responsive to operation of said manipulating, emitting code that calls said public functions and/or data attributes in said first language to obtain called public functions and/or data attributes and converts said called public functions and/or data attributes to representations of said called public functions and/or data attributes formed in a different computer language compatible with said standardized software architecture.

Same rejection as referred to the rationale addressed in As per Claim 1.

As per Claim 25: Chen discloses, *The method of claim 24 further comprising forwarding said representations to desired destinations within and beyond said system.*

Same rejection as referred to the rationale addressed in As per Claim 2.

As per Claim 26: Chen discloses, *The method of claim 25 and wherein said first computer language is RAID++ and said different computer language is XML/CIM.*

Same rejection as referred to the rationale addressed in As per Claim 3.

As per Claim 27: Chen discloses, *The method of claim 25 and wherein said first computer language is an object-oriented language defining computer data and commands as objects, said manipulating comprising:*

opening one of said header files containing a declaration of one of said objects (Refer to the rationale addressed to the means for manipulating in As per claim 1);

parsing said one of said header files to obtain name of class and name of parent class to which said one of said objects belongs (See col. 689, sec. 4.1); *and,*

creating a subroutine for accepting said one of said objects in said first computer language and generating the equivalent of said one of said objects in a different computer language compatible with said standardized software architecture (See p. 683, left col., A C++ DATA MODEL; and see p. 690, left col. Sec. B, 'using an instance compiler that takes a specification file for a new language or document type and generates the complete query and visualization environment automatically').

As per Claim 28: Chen discloses, *The method of claim 27 further comprising inhibiting initiation of operation of said converting until said public functions and/or data attributes of said header files are located.*

Same rejection as referred to the rationale addressed in As per Claim 2.

As per Claim 29: Chen discloses, *The method of claim 28 further comprising initiating operation of said converting upon locating the first of any one of said public functions and/or data attributes.*

(See p. 683, left col., A C++ DATA MODEL; and see p. 690, left col. Sec. B, 'using an instance compiler').

As per Claim 30: Chen discloses, *The method of claim 29 and wherein said first computer language is C++ and said different computer language is XML/CIM.*

Same rejection as referred to the rationale addressed in As per Claim 7.

As per Claim 31: Chen discloses, *The method of claim 27 and wherein said first computer language is a first object-oriented language capable of pictorial representation typically in a parent child tree configuration and said different computer language is a second object-oriented language capable of pictorial representation typically in a flat database configuration.* Same rejection as referred to the rationale addressed in As per Claim 8.

As per Claim 32: Chen discloses, *The method of claim 24 further comprising inhibiting initiation of operation of said converting until said all public function and data attributes of at least one of said header files are located.* Same rejection as referred to the rationale addressed in As per Claim 9.

As per Claim 33: Chen discloses, *The method of claim 31 further comprising inhibiting initiation of operation of said converting until said public functions and/or and data attributes of at least one of said header files are located.* Same rejection as referred to the rationale addressed in As per Claim 9.

As per Claim 34: Chen discloses, *The method of claim 24 and wherein said standardized software architecture is preferred non-legacy software architecture.* (Refer object model visualized HTML, XML, and CIAO as discussed in section B, p. 690, and sec. VI, p. 692).

As per Claim 35: Chen discloses, *The method of claim 24 and wherein said management software is storage management software.* Same rejection as referred to the rationale addressed in As per Claim 10.

As per Claim 36: Chen discloses, *The method of claim 24 and wherein said management software is selected from the group consisting of storage, printer, and server management software.* Same rejection as referred to the rationale addressed in As per Claim 11.

As per Claim 37: Chen discloses, *A computer program product including management software written in a first language and embodied in a computer-readable medium for operation on a computer system designed in accordance with first software architecture and not compatible with other than said first software architecture, said computer program product comprising: programmable code for utilizing a schema having header files in said first language; programmable code for manipulating said header files to locate public functions and/or data attributes of said header files; and, programmable code, responsive to said manipulating, for emitting special code that calls said public functions and/or data attributes in said*

first language to obtain called public functions and/or data attributes and converts said called public

functions and/or data attributes to representations of said called public functions and/or data attributes

formed in a different computer language compatible with said other than said first software architecture.

Same rejection as referred to the rationale addressed in As per Claim 1.

As per Claim 38: Chen discloses, *The computer program product of claim 37 further comprising*

programmable code for forwarding said representations to desired destinations within and beyond said

system. Same rejection as referred to the rationale addressed in As per Claim 2.

As per Claim 39: Chen discloses, *The computer program product of claim 38 and wherein said first*

computer language is an object-oriented language defining computer data and commands as objects,

said programmable code for manipulating comprising: programmable code for opening one of said

header files containing a declaration of one of said objects;

programmable code for parsing said one of said header files to obtain name of class and name of parent

class to which said one of said objects belongs; and, programmable code for creating a subroutine for

accepting said one of said objects in said first computer language and generating the equivalent of said

one of said objects in a different computer language compatible with said other than said first software

architecture.

Same rejection as referred to the rationale addressed in As per Claim 4.

As per Claim 40: Chen discloses, *The computer program product of claim 39 and wherein said first*

computer language is RAID++ and said different computer language is XML/CIM. Same rejection as

referred to the rationale addressed in As per Claim 3.

As per Claim 41: Chen discloses, *The computer program product of claim 40 further comprising*

programmable code for inhibiting initiation of operation of said programmable code for converting until

said public functions and/or data attributes of said header files are located. Same rejection as referred to

the rationale addressed in As per Claim 5.

As per Claim 42: Chen discloses, *The computer program product of claim 40 further comprising*

programmable code for initiating operation of said programmable code for converting upon locating the

first of any one of said public functions and/or data attributes. Same rejection as referred to the rationale

addressed in As per Claim 6.

As per Claim 43: Chen discloses, *The computer program product of claim 39 and wherein said first*

computer language is C++ and said different computer language is XML/CIM. Same rejection as referred

to the rationale addressed in As per Claim 7.

As per Claim 44: Chen discloses, *The computer program product of claim 37 and wherein said first*

computer language is a first object-oriented language capable of pictorial representation typically in a

parent-child tree configuration and said different computer language is a second object-oriented language

capable of pictorial representation typically in a flat database configuration. Same rejection as referred to

the rationale addressed in As per Claim 8.

As per Claim 45: Chen discloses, *The computer program product of claim 44 further comprising*

programmable code for inhibiting initiation of operation of said programmable code for converting until

said public functions and/or and data attributes of at least one of said header files are located. Same

rejection as referred to the rationale addressed in As per Claim 9.

As per Claim 46: Chen discloses, *The computer program product of claim 37 and wherein said first*

software architecture is legacy software architecture and said other than said first software architecture is

preferred non-legacy software architecture. Refer to C++ and C++ Object Model visualized by

HTML/XML.

As per Claim 47: Chen discloses, *The computer program product of claim 37 and wherein said*

management software is storage management software. Same rejection as referred to the rationale

addressed in As per Claim 10.

As per Claim 48: Chen discloses, *The computer program product of claim 37 and wherein said*

management software is selected from the group consisting of storage, printer, and server management

software. Same rejection as referred to the rationale addressed in As per Claim 11.

As per Claim 49: Chen discloses, *A computer program product compatible with preferred non-legacy*

software architectures and operating in a computer system employing management software written in a

first computer language compatible with legacy software architecture and not compatible with said

preferred non-legacy software architectures, said computer program product embodied in a computer

readable medium and comprising:

programmable code for utilizing a schema having header files in said first language;

programmable code for manipulating said header files to locate public functions and/or data

attributes of said header files;

programmable code, responsive to said manipulating, for emitting special code that calls said

public functions and/or data attributes in said first language to obtain called public functions and/or data

attributes; and,

programmable code for converting said called public functions and/or data attributes to

representations of said called public functions and/or data attributes formed in a plurality of different

computer languages each being compatible with at least one of said preferred non-legacy software

architectures.

Same rejection as referred to the rationale addressed in As per Claim 1.

As per Claim 50: Chen discloses, *The computer program product of claim 49 and wherein said*

management software is storage management software. Same rejection as referred to the rationale

addressed in As per Claim 10.

As per Claim 51: Chen discloses, *The computer program product of claim 49 and wherein said*

management software is selected from the group consisting of storage, printer, and server management

software. Same rejection as referred to the rationale addressed in As per Claim 11.

As per Claim 52: Chen discloses, *In a computer network including a computer system having a functional*

system therein with management software including a schema for managing said functional system under

control of said computer system in accordance with first software architecture, a translator-compiler for

permitting communication about said managing said functional system to be transmitted between said

computer system and computer devices operating under second software architecture, said translator

compiler comprising:

program code for accessing header files within said schema to obtain a header file containing

particular information;

program code for parsing said header file to obtain a particular result;

program code for opening an output file for storage of other than said particular information

related to said particular result;

program code for continued parsing of said header file to locate public functions and/or data

attributes; and,

program code for emitting special code to said output file that calls said public functions and/or

data attributes to obtain called public functions and/or data attributes and for converting said called public

functions and/or data attributes to language compatible with said second software architecture;

whereby communication about managing said functional system transmitted between said

computer system and said computer devices operating under said second software architecture is

obtained. See rationales addressed in As Per Claim 1.

<u>As per Claim 53</u>: Chen discloses, *In the computer network of claim 52 and wherein said functional*

system is a storage system and said management software is storage management software. Same

rejection as referred to the rationale addressed in As per Claim 10.

<u>As per Claim 54</u>: Chen discloses, *In the computer network of claim 52 and wherein said functional system*

is a SAN and said management software is SAN management software. Same rejection as referred to the

rationale addressed in As per Claim 11.

<u>As per Claim 55</u>: Chen discloses, *In the computer network of claim 52 and wherein said first software*

architecture is legacy software architecture and said second software architecture is non-legacy software

architecture. Refer to C++, and C++ Object Model, specification file.

<u>As per Claim 56</u>: Chen discloses, *In the computer network of claim 52 and wherein said functional system*

is selected from the group consisting of storage system, printer system, server system or other-

component system and said management software is selected from the group consisting of storage

management software, printer management software, server management software and other-component

management software respectively. Same rejection as referred to the rationale addressed in As per Claim

11.

As per Claim 57: Chen discloses, *In the computer network of claim 52 and wherein said computer devices are located within said network.* Refer to any computer.

As per Claim 58: Chen discloses, *In the computer network of claim 52 grid wherein said computer devices are located outside of said network.* Refer to any computer.

As per Claim 59: Chen discloses, *In a computer network including a computer system and a functional system controlled by said computer system, management software compatible with legacy software architecture having header files, said management software being deployed on both said computer system and said functional system, said management software comprising:*

translator software means for receiving and manipulating said header files;

software means for receiving first requests in first language incompatible with said legacy software architecture;

software means responsive to operation of said translator software means for obtaining responses to said first requests in: second language compatible with said legacy software architecture; and,

software means for converting said responses to equivalent responses compatible with said first language and for communicating said equivalent responses to the destination from which, or to destinations related to that from which, said first requests originated.

See rationales addressed in As Per Claim 1.

As per Claim 60: Chen discloses, *In the computer network of claim 59 and wherein said functional system is a storage system.* Refer to database.

As per Claim 61: Chen discloses, *In the computer network of claim 59 and wherein said functional system is a SAN.* Refer to relational database.

As per Claim 62: Chen discloses, *In the computer network of claim 59 and wherein said first requests are received from outside of said network.* See sec VI, p. 692.

As per Claim 63: Chen discloses, *In a computer system compatible with computer software architecture, a computer-readable medium containing management software for controlling at least one processor in said system to perform a method of responding to requests, said method comprising:*

receiving first requests in first language incompatible with said computer software architecture;

obtaining responses to said first requests in second language compatible with said computer software architecture; and,

converting said responses to equivalent responses compatible with said first language and communicating said equivalent responses to the destination from which, or to destinations related to that from which, said first requests originated.

See rationales addressed in As Per Claim 1, and particularly see sec. VI, Web-based/Web Service.

As per Claim 64: Chen discloses, *In the computer system of claim 63 and wherein said computer software architecture is legacy software architecture.* Refer to C++ software.

As per Claim 65: Chen discloses, *In the computer system of claim 64 and wherein said management software is storage management software.* See rationales addressed in As Per Claim 10.

As per Claim 66: Chen discloses, *In the computer system of claim 64 and wherein said management software is SAN management software.* See rationales addressed in As Per Claim 10.

As per Claim 67: Chen discloses, *In the computer system of claim 66 and wherein said destination is located outside of said computer system.* See Sec. VI.

As per Claim 68: Chen discloses, *A computer program product embodied in a computer readable medium to be operated on a computer compatible with computer software architecture comprising:*

programmable code for receiving first requests in first language incompatible with said computer software architecture;

programmable code for obtaining responses to said first requests in second language compatible with said computer software architecture; and,

programmable code for converting said responses to equivalent responses compatible with said first language and for communicating said equivalent responses to the destination from which, or to destinations related to that from which, said first requests originated.

See rationales addressed in As Per Claim 1.

As per Claim 69: Chen discloses, *The computer program product of claim 68 and wherein said computer software architecture is legacy software architecture.* Refer to C++.

<u>As per Claim 70</u>: Chen discloses, *A method for managing functional systems to be practiced on a computer compatible with computer software architecture comprising:*

receiving first requests in first language incompatible with said computer software architecture;

obtaining responses to said first requests in second language compatible with said computer software architecture; and,

converting said responses to equivalent responses compatible with said first language and communicating said equivalent responses to the destination from which, or to destinations related to that from which, said first requests originated.

See rationale addressed in As per Claim 1, and further relate to sec B, p. 690, and sec. VI, p. 692.

<u>As per Claim 71</u>: Chen discloses, *The method of claim 70 and wherein said computer software architecture is legacy software architecture.* Refer to C++.

<u>As per Claim 72</u>: Chen discloses, *The method of claim 71 and wherein said functional systems include a storage system.* See Relational Database.

<u>As per Claim 73</u>: Chen discloses, *The method of claim 71 and wherein said functional systems include a SAN.* See Relational Database.

<u>As per Claim 74</u>: Chen discloses, *The method of claim 71 and wherein said functional systems are selected from the group consisting of storage systems, printer systems, and server systems.* See rationale addressed in As per Claim 11.

<u>As per Claim 75</u>: Chen discloses, *In a computer network including a computer system and a storage system controlled by said computer system, a method for managing storage compatible with software architecture having header files, said method being deployed on both said computer system and said storage system, said method comprising:*

translating and manipulating said header files to obtain translated and manipulated header files;

receiving first requests from outside of said network in first language incompatible with said software architecture;

in cooperation with said translated, and manipulated header files, obtaining responses to said first requests in second language compatible with said software architecture; and,

in cooperation with said translated and manipulated header files, converting said responses to equivalent responses compatible with said first language and communicating said equivalent responses to said outside of said network.

See rationale addressed in As per Claim 1, and further relate to sec B, p. 690, and sec. VI, p. 692.

As per Claim 76: Chen discloses, *The method of claim 75 and wherein said software architecture is legacy software architecture.* Refer to C++.

As per Claim 77: Chen discloses, *In the computer network of claim 76 further comprising said storage system is a SAN.* Refer to relational database.

As per Claim 78: Chen discloses, *In the computer network of claim 75 and wherein said first language is a first object-oriented language capable of pictorial representation typically in a flat database configuration and said second language is a second object-oriented language capable of pictorial representation typically in a parent-child tree configuration.* Same rejection is as referred to the rationale addressed in As per Claim 8.

As per Claim 79: Chen discloses, *In the computer network of claim 78 and wherein said first language is CIM/XML and said second language is C++.* Similar rejection is as referred to the rationale addressed in As per Claim 8.

As per Claim 80: Chen discloses, *In the computer network of claim 79 and wherein said C++ language is RAID++.* Similar as C++, rejection is as referred to the rationale addressed in As per Claim 8.

As per Claim 81: Chen discloses, *In an improved network including a first computer network operating in accordance with first: software architecture and a second computer network operating in accordance with second software architecture, the improvement comprising:*

an interface between said first computer network and said second computer network to automatically convert communication from said second computer network into a form compatible with said first computer network, and to automatically convert response to said communication generated by said first computer network into a form compatible with said second computer network.

See rationale addressed in As per Claim 1, and further relate to sec B, p. 690, and sec. VI, p. 692.

As per Claim 82: Chen discloses, *In the improved network of claim 81 and wherein said first software architecture is legacy software architecture and said second software architecture is non-legacy software architecture.* Refer to C++, and C++ Object model/specification file visualized by XML.

As per Claim 83: Chen discloses, *The improvement of claim 82 and wherein said first computer network operates in accordance with said legacy software architecture supporting a first object-oriented computer language capable of pictorial representation typically in a parent-child tree configuration, and wherein said second computer network operates in accordance with said non-legacy software architecture supporting a second object-oriented computer language capable of pictorial representation typically in a flat database configuration.* See rationale addressed in As Per Claim 8.

As per Claim 84: Chen discloses, *The improvement of claim 83 and wherein said first object-oriented computer language is C++ and wherein said second object-oriented computer language is XML/CIM.* See rationale addressed in As Per Claim 7.

As per Claim 85: Chen discloses, *The improvement of claim 84 and wherein said communication includes management software communication.* Refer to relational database and further relate sec. VI, p. 692.

As per Claim 86: Chen discloses, *The improvement of claim 85 and wherein said management software communication includes storage management software communication.* See rationale addressed in As per Claim 10.

As per Claim 87: Chen discloses, *The improvement of claim 86 and wherein said storage management software communication relates to SAN communication.* See rationale addressed in As per Claim 10.

As per Claim 88: Chen discloses, *The improvement of claim 85 and wherein said management software communication includes storage, printer, and server communications.* See rationale addressed in As per Claim 11.

As per Claim 89: Chen discloses, *The improvement of claim 81 and wherein said response is communicated to the destination from which, or to destinations related to that from which, said communication originated.* See sec. VI.

Conclusion

5.   Any inquiry concerning this communication or earlier communications from the examiner should be directed to Ted T. Vo whose telephone number is (571) 272-3706. The examiner can normally be reached on 8:00AM to 5:30PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Wei Y. Zhen can be reached on (571) 272-3708.

The facsimile number for the organization where this application or proceeding is assigned is the Central Facsimile number **571-273-8300**.

Any inquiry of a general nature or relating to the status of this application should be directed to the TC 2100 Group receptionist: 571-272-2100. Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Ted T. Vo
Primary Examiner
Art Unit 2191
December 09, 2005